

AMENDMENTS TO THE SPECIFICATION

Please amend the following paragraphs of the Specification without prejudice or disclaimer to read as follows:

In the paragraph spanning pages 1 and 2:

With reference to Figure 1, a typical computer system 100 includes hardware 102 such as a processor, network interface, a keyboard, a mouse, a monitor and the like. Applications 108 suitably communicate with the hardware to perform various functions through operating system 106. If applications 108 include network functionality, a network interface 104 typically receives network calls from the application 108 via operating system 106 and relays them as appropriate to network hardware included with hardware 108. An exemplary network interface includes the network design interface specification (NDIS) available from the Microsoft™ Corporation of Redmond, Washington, although of course other network interfaces could be used in conjunction with various operating systems and hardware configurations.

In the second full paragraph on page 2:

Although various administration tools have existed in the past, these tools have exhibited one or more marked disadvantages. The Intellimirror™ product available from the Microsoft Corporation of Redmond, Washington, for example, operates at the application level 108. Similarly, the Norton Utilities™ and Norton Ghost™ programs available from the Symantec Corporation of Cupertino, California operate at application level 108. Because these programs operate at the application level, their proper execution requires that hardware 102, network layer 104 and operating system 106 be available and functional. If any of these layers 102, 104 or 106 are damaged or otherwise unavailable, the administrative program will not typically function properly.

In the first full paragraph on page 6:

To simplify the description of the exemplary embodiments, the invention is frequently described as pertaining to a system of administering personal computers. It will be appreciated, however, that many applications of the present invention could be formulated. For example, the present invention could be used to administer, upgrade, monitor, configure or control any type of computer running any operating system such as any version of Windows™, MacOS™, BeOS™, Linux™, UNIX™, or the like. Similarly, the invention could be used in conjunction with any type of personal computer, network computer, workstation, minicomputer, mainframe, or the like. Moreover, although the invention is frequently described herein as being implemented with TCP/IP communications protocols, it will be readily understood that the invention could also be implemented using IPX™, Appletalk™, IP-6, NetBIOS™, OSI or any number of existing or future protocols. Additionally, the PXE pre-boot environment discussed herein could be replaced with any other pre-boot scheme, including any aspect of the Wired for Management™ (WFM) baseline or any other pre-boot sequence. Moreover, the database applications described herein may be implemented with any type of directory services application, relational database, object-oriented database, hierarchical database, data structure or the like, as described more fully below.

In the paragraph spanning pages 8 and 9:

With continued reference to Figure 2, client computers 202 may be any type of computer system (or combination of computer system types) that are to be remotely managed. In various embodiments, client computers 202 include memory, a processor and a hard drive (not shown in Figure 2). Each hard drive may include an individually-bootable operating system such as Windows 95™, Windows 2000™, Windows NT™, any other version of Windows™, Linux™, UNIX™, MacOS™ or the like. Alternatively, one or more client computers 202 may be configured as "network computers" that obtain their operating system from a network server 206, or another server not shown in Figure 2. Each client computer 202 is coupled to data network 210 via any appropriate device or technique. In an exemplary embodiment, client computers 202 include network interface cards (NICs) that facilitate communications between client computer 202 and various network hosts. Such NICs may include media access control (MAC) addresses stored in hardware, software or firmware that uniquely identify the particular NIC. Such MAC addresses may be assigned and configured by NIC manufacturers such as the 3Com™ corporation, the Intel™ Corporation and others. In various embodiments, each NIC also includes remote boot functionality in hardware, software or firmware as described below. An example of remote boot functionality is the pre-boot execution environment (PXE)™ formulated by the Intel Corporation of Santa Clara, California and described in "Pre-boot Execution Environment (PXE) Specification Version 2.0" dated December 28, 1998, which is incorporated herein in its entirety by reference.

In the first full paragraph on page 9:

Network 210 may be any device or technique capable of transmitting data between computers such as clients 202 and server 206. In various embodiments, network 210 is an Ethernet™, token ring or fiber optic network, although it will be appreciated that any suitable cabling, networking, or data transmission technique (such as any form of wireless networking) could be used.

In the second full paragraph on page 9:

Server computer 206 (also referred to simply as a "server") may be any computer or data processing system that is used to administer the files, data and/or usage of client computers 202. In various embodiments, server 206 is a personal computer or workstation running, for example, the LINUX™, UNIX™, Windows NT™, Windows 2000™ or any other operating system. In other embodiments, server 206 is a personal computer configured as a network server using, for example, Netware Server™ version 4.0 (or greater) software available from the Novell™ corporation of Provo, Utah.

In the paragraph spanning pages 9 and 10:

Server 206 may maintain a database 208 of characteristics relating to client computers 202, as appropriate. Database 208 may be implemented with any hierarchical or relational database, registry or directory services product that is capable of storing workstation attributes. Examples of products that could be used to implement database 208 include the Netware Directory Services™ (NDS) directory available from the Novell corporation of Provo, Utah, or the Microsoft Active Directory™ available from the Microsoft Corporation of Redmond, Washington. Various standards relating to storing information in directory services

include X.500, the lightweight directory access protocol (LDAP), the directory access protocol (DAP), NDS and AD. Other embodiments may incorporate database software available from, for example, Netscape™, Sybase™, Oracle™, IBM™ or Microsoft™. As used herein, the term "database" refers to any database, database management, directory services or similar application.

In the paragraph spanning pages 11 and 12:

Two exemplary embodiments of a server application 300 are discussed herein. In the first embodiment (shown in Figure 9), various interface windows are provided with configuration options for various client computers 202. The options may be selected by an administrator as appropriate. In the second embodiment (represented by Figures 3 and 10), interface windows are provided that allow an administrator to associate scripts with events, which are in turn associated with templates that correspond to particular client computers 202. In various embodiments, the interface windows may be generated as Netware Snap-in™ applications for Novell's Netware Administrator™ programs, or as snap-ins for the Microsoft Management Console™ (MMC) application, for example. It will be understood that the interfaces shown in the various figures are exemplary and that the functionality of server 206 could be implemented with any type of graphical, textual or other form of interface. Further, the embodiments of server application 300 described herein are meant to be illustrative, and of course various elements of the two embodiments may be combined or eliminated as appropriate without departing from the scope of the invention.

In the paragraph spanning pages 13 and 14:

With reference now to Figure 3, a second exemplary embodiment of server application 300 suitably associates configuration scripts 302 with events 304, which in turn may be associated with individual computer templates 306, which in turn may be manually or automatically associated with computer objects 308 and/or computer group objects 310. In various embodiments scripts 302, events 304, templates 306, computers 308 and computer groups 310 may be represented as objects or other data constructs, and may be stored in database 208 and/or in directory services. Various graphical techniques could be used to implement the various object associations. In an exemplary embodiment, the various objects may be represented as objects or other elements in the Netware Directory Services™ or Active Directory™ administration tree and may be managed with the Netware Administrator™ program or Microsoft Management Console™ (MMC).

In the paragraph spanning pages 15 and 16:

Computer templates (also referred to as computer template objects) 306 may be used to group events together, as described below, and to associate the various types of objects with physical computing devices (e.g. client computer 202). More particularly, template objects 306 may be used to associate a client computer's hardware properties to a database object (such as a Netware Directory Services™ or Microsoft Active Directory™ object), and to associate that client computer 202 to the processes invoked during the pre-boot management process. To continue the example presented above, if "flash BIOS" script 302 and "Next Boot" event 304 were associated with a particular template 306, template 306 would indicate which client computer 202 would have its BIOS flashed at the next boot of the machine. Each template 306 may be associated with multiple events 304 and scripts 306. In various embodiments, default templates 306 may be

created that are associated with client computers 202 until a personalized template can be specified. The default template 306 may be associated with scripts to execute standard instructions on all client computers 202 as they are booted for the first time within the system, for example. In such embodiments, one of the instructions in the scripts associated with the default templates might be to interrogate the machine as to its type, operating system, hardware configuration, etc. as described below.

In the first full paragraph on page 16:

A copy of the default template 306 may be created and suitably modified for each client 202, or new templates 306 may be created from other templates or from scratch as appropriate. Templates 306 may be created or modified by, for example, using an administration program such as the Network Administrator™ program (such as version 5.1 or later) available from the Novell™ corporation, the Microsoft MMC™ or Enterprise Manager™ program available in conjunction with the Active Directory™ product from the Microsoft™ corporation, or any other administration program. Alternatively, templates 306 may be modified by directly opening and editing the contents of the template with a text editor or other program.

In the paragraph spanning pages 16 and 17:

Computer objects 308 suitably store physical attributes (such as hardware and firmware settings) of client computers 202. Computer objects 308 may be automatically created upon discovery of a new client computer 202 (e.g. when a client computer 202 sends a first PXE or other boot request to server 206), and may be indexed in database 208 by the unique hexadecimal string associated with each client computer's MAC address, or according to any other indexing scheme. Computer objects 308 may be manually associated with template objects 306 (for example by manually selecting a particular computer object indicator in the Novell Network Administrator™ or Microsoft MMC™ program, or another similar program). Templates 306 may also be automatically associated with a computer object 308 according to, for example, hardware or firmware attributes of a client computer 202. As described more fully above, attributes of a particular client 202 can be determined automatically, and these hardware (or other) attributes may be used to match the particular client computer 202 with an appropriate template 306 within the directory services/database application 308.

In the second full paragraph on page 17:

Referring now to Figures 10-12, exemplary user interfaces for configuring server application 300 are shown. As stated above, each of the interface windows may be created as, for example, Novell Netware™ or Microsoft MMC™ snaps, or according to any other format or technique.

In the paragraph spanning pages 20-21:

A network address server (such as server 206) may recognize client 202 by, for example, the MAC address associated with client 202, and may respond with an appropriate response message 312, which may include a client network address, as well as router information, a subnet mask and the like. In various embodiments, the client network address is an internet protocol (IP) address or a Netware™ (IPX™) address, although any network addressing technique could be used.

In the second full paragraph on page 26:

With reference now to Figure 5, various embodiments of exemplary client software (which may be provided with boot image 500) may include a bootable operating system (OS) 502 such as DOS™, LINUX™, UNIX™ or the like. Operating system 502 may include a small operating system kernel (such as a DOS™ or LINUX™ kernel), a command processor, an extended memory manager and a command interpreter 506. In an exemplary embodiment, operating system 502 automatically executes command interpreter 506 after the kernel is loaded.

In the paragraph spanning pages 32-33:

With momentary reference again to Figures 4A and 5, registry/directory administration program 410 may be suitably used to administer the registry associated with the operating system stored on a hard disk or other storage device affiliated with client computer 202. The Windows 95™, 2000™ and NT™ operating systems, for example, maintain a registry of hardware configurations, software files, and the like. This registry may be stored as a file on client computer 202. Registry administration program 410 suitably compares the registry to a duplicate registry stored on server 206 or on client computer 202 to determine any configuration errors or to adjust configuration parameters. Alternatively, registry administration program 410 may identify registry files or keys used by client computer 202 by, for example, checking an oemsetup.inf file. Keys identified in the file can be compared with, for example, keys required by the hardware on client 202 identified by the preloader. The registry itself can be scanned and any keys that are missing can be added, repaired, or replaced based upon the key data identified.